

13  
150 pts.

# PRUM

Enciclopedia Práctica del Spectrum



Nueva Lente/Ingelek





# FUNCIONES MATEMÁTICAS



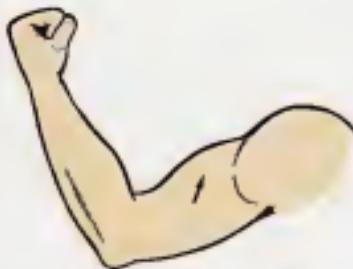
AJÚD esta denominación general, trataremos las funciones de potenciación, exponentiación, logarítmicas y trigonométricas. La estructura de esas funciones es muy similar a las de tratamiento numérico que vimos anteriormente, no obstante, su uso es bastante menos frecuente, al resumirse su aplicación al campo científico o de cálculo matemático.

Por ello, no debemos asustarnos si en algún momento no llegamos a comprender exactamente las explicaciones sobre la utilización de estas funciones. Tengamos en cuenta que, si realmente no conocemos su uso, nunca las utilizaremos en nuestros propios programas. Por tanto, nos bastará con saber qué son, así como su sentido, para facilitarnos la comprensión e introducción de listados de programas, confeccionados por otras personas que sí las utilizan.

Afí pues, los poco introducidos en el campo de las matemáticas, no deben preocuparse por no comprender completamente las explicaciones que a continuación se brindan. En todo caso, se ha procurado simplificar estas en lo posible, lo que pueda dar lugar a algunas imprecisiones que no son dignas de ser tenidas en consideración.

## POTENCIACIÓN

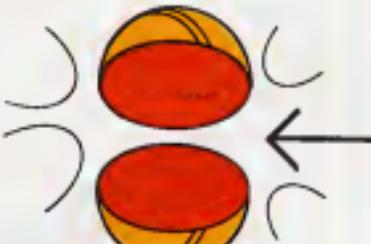
Es de todos conocido que la potenciación consiste en multiplicar un número, denominado base, por sí mismo, tantas veces como indica el superíndice que le sucede, que se denomina exponente. Así por ejemplo, el término  $4^2$ , que se lee "cuatro elevado a dos" (o "cuatro al cuadrado"), tiene como resultado  $4 \times 4 = 16$ . Lógicamente, debido a la restricción de no poder utilizar la notación habitual de superficies, en BASIC se adoptó el criterio de representar esa operación mediante la base seguida de un signo "Recha hacia arriba" (↑) y, a continuación, el exponente. Con lo cual, la operación del ejemplo anterior se escribiría en BASIC de la siguiente manera:  $4\uparrow 2$ .



Para la obtención de la función de potenciación, en el Spectrum se utiliza un carácter ejemplificado arriba.

Dentro de las funciones matemáticas, probablemente sea ésta la de uso más frecuente, sobre todo con el exponente dos. Daremos un breve repaso a nuestros conocimientos básicos sobre esta operación.

La potenciación, al igual que el resto de las operaciones matemáticas que hemos visto hasta el momento (suma, multiplicación, etc.), tiene una operación inversa la cual nos permite conocer qué base elevada a un exponente proporciona determinado resultado; esta operación se denomina radicación, y se representa por una uve (V) en cuya parte superior se expresa el índice de la raíz, unida a una línea de super-rayado debajo de la cual se escribe el radicando (argumento de la función).



La potenciación aduce exponentes fraccionarios, mediante los cuales conseguiremos la función de radicación.

!

Le funciones exponencial y logarítmica son inversas.



Para la obtención en el Spectrum de logaritmos en una base distinta de e, tenemos que utilizar una proporción de los logaritmos, más por la cual el logaritmo en base B de N, es igual al exponente del logaritmo natural de N, dividido por el logaritmo natural de la base B.

!

Los argumentos de las funciones trigonométricas deben expresarse en radianes.

\*

La potenciación consiste en multiplicar un número (base) por la misma, tantas veces como indique el exponente o índice (exponente) que lo sigue.

En el caso concreto del ejemplo anterior, la operación inversa al cuadrado de cuadro, es la raíz cuadrada de decimal, representada  $\sqrt{16}$ , cuyo resultado es 4. Así pues, se cumple que dada una base cualquiera  $B$ , y un exponente cualquiera  $E$ :

$$B^E = X \quad B = \sqrt[E]{X}$$

Pero el igual que no es posible representar en la pantalla la potenciación con su simbología tradicional, menos aún podemos utilizar la notación horagonal para la radicación. Así pues, ¿dónde representa en BASIC esta operación? En el caso concreto de la raíz cuadrada (índice dos), que es sin duda el cálculo de radicación más común, se utiliza una función específica: SQR. Esta es la abreviatura de la palabra inglesa SQuare, que significa CUADRADO. Por tanto, si efectuamos PRINT SQR 16 obtendremos el resultado cuatro [4].

Ahora bien, tanto la potenciación como la radicación se pueden efectuar con cualquier exponente o índice. En el caso de la primera operación, expresar un exponente distinto a dos, es bien sencillo, no hay más que cambiar el número escrito tras la flecha por el exponente que deseemos, sin embargo, en el caso de la radicación, la función BASIC SQR implica una raíz cuadrada, ¿cómo expresar entonces un índice distinto de dos? Para responder a esta pregunta es preciso disponer de una cierta base matemática.

Los conocimientos de cálculo matemático de nuestro Spectrum son considerables, y prácticamente es seguro que, sin ánimo de monopolio, superen los que nosotros tenemos. Por ello, la forma en la que el ordenador realiza determinadas operaciones internamente, no se corresponde exactamente con la manera tradicional porque las llevaremos a cabo los profesores en la materia, aunque el sistema empleado sea más, o al menos igualmente, eficaz. Un caso en el que se cumple esta afirmación es el de la potenciación:



Para hallar la raíz n-ésima de  $X$ , debemos elevar  $X$  a una potencia por  $N$ .

La función logarítmica en el Spectrum emplea la base e (logaritmo natural), y se obtiene por LN.



Como ya hemos dicho, elevar una base a un exponente igual a multiplicar dicha base por sí misma, tantas veces como indique el exponente; sin embargo, esto no es la forma en la que el Spectrum lleva a cabo dicha operación. Nuestro ordenador se basa en su capacidad para realizar la operación matemática denominada LOGARÍTMICO, que dentro de muy poco pasaremos a estudiar. Más concretamente, el sistema de potenciación se apoya en una propiedad de los logaritmos, por la cual la base  $B$  elevada al exponente  $E$ , es equivalente al antilogaritmo del exponente  $E$  multiplicado por el logaritmo de la base. Esto se expresa en BASIC de la siguiente manera: B^E=EXP (E\*LN B)

Según este sistema para el cálculo de la potencia tiene una ventaja claramente la posibilidad de elevar números a exponentes fraccionarios. evidentemente, con el método de productos sucesivos, esta operación se haría bastante complicada, porque podríamos multiplicar un número por sí mismo dos veces, tres, cuatro... pero ¡jáma hacer para multiplicarlo por sí mismo 3.735 veces!





Una buena regla mnemotécnica para recordar el valor de  $e$ , la proporciona el hechizo de que su aproximación coincide con 2.7, seguido del resto de la muestra de Goya (1.628).

La pregunta que se plantea inmediatamente, se refiere a la utilidad de elevar una base a un exponente fraccionario; pues bien, es mucha, y pronto lo comprobaremos, puesto que nos permitirá obtener las raíces de cualquier índice a través de la potenciación.

El hecho antes mencionado, se fundamenta en una propiedad de las potencias, por la cual una base cualquiera  $B$ , elevada a un exponente fraccionario  $N/D$ , es igual a la raíz de índice  $D$  de la base  $B$  elevada a  $N$ , o lo que es lo mismo

$$B^{N/D} = \sqrt[D]{B^N}$$

Por tanto, puesto que cualquier base elevada a exponente 1 tiene por resultado la misma base, podemos particularizar la expresión anterior, asegurando que la raíz  $n$ -ésima de cualquier número se igual al número elevado a uno dividido por el índice. Esto quiere decir, que la obtención de la raíz cuadrada se puede llevar a cabo de dos formas diferentes: una por medio de la función BASIC específica (SQR) y otra, elevando la base a un medio (1/2). Gracias a este sistema, podemos obtener cualquier raíz, así por ejemplo, la raíz del índice tres de 343 se puede averiguar mediante la siguiente instrucción. PRINT 343^(1/3)

Para finalizar con el tema de las potencias, y puesto que hemos visto como obtener las radicaciones mediante esta operación, vamos a enumerar algunas propiedades interesantes, que no están directamente relacionadas con el BASIC, pero que nos pueden ser de utilidad a la hora de realizar algún programa que precise esta operación:

$$\begin{aligned} B^0 &= 1 \\ B^{-E} &= (1/B)^+E \\ B^E \cdot B^F &= B^{E+F} \\ B^E / B^F &= B^{E-F} \end{aligned}$$

En cuanto a la prioridad en el orden de ejecución la potenciación tiene la más alta entre las operaciones matemáticas elementales. Otra precaución a tener para la correcta ejecución de las potenciaciones, es recordar que la utilización de bases negativas produce un error del tipo A Invalid argument, el igual que ocurre con la función SQR, como tuvimos oportunidad de comprobar en el capítulo anterior cuando estudiámos la función SQRT.



## FUNCIONES LOGARÍTMICAS

i!

Comenzaremos este epígrafe definiendo la función que trata el LOGARITMO en cualquier base. Si de un número  $N$ , es el exponente al que se ha de elevar  $B$  para obtener  $N$ . Se trata pues de una de las operaciones inversas a la potenciación, por tanto, hemos de tener cuidado de no intentar efectuar el logaritmo de un número negativo para no obtener de nuevo el desagradable mensaje de A Invalid argument (argumento inválido). Los logaritmos, como hemos visto, se pueden expresar en cualquier base, pero sin duda son las más ampliamente utilizadas en el cálculo matemático, la base 10 y la base e. Los de la base 10, son también conocidos bajo el nombre de los logaritmos decimales, vulgares o de BRIGGS, por ser este matemático el que confeccionó la primera tabla de logaritmos de este tipo.



La función de potencia (EXP), permite utilizar exponentes fraccionarios, gracias a lo cual facilita la obtención de raíces cuadradas.

\*

Algunos de los principales de las potencias que conviene recordar son:  $A^0=1$ ,  $A^{-E}=1/A^E$ ,  $A^{1/B}=1/A^B$ ,  $A^B \cdot C=A^{B+C}$ ,  $A^B/C=A^{B-C}$

\*

La fórmula general para la obtención de la raíz n-ésima de  $B$  es  $B^{1/N}$ .

!

Los logaritmos obtenidos directamente por el Spectrum son los del tipo neperiano o natural (base e).



El Spectrum incopora la constante PI internamente, con un valor aproximado 3.1415927.



La función exponencial (antilogaritmo) se obtiene en el Spectrum mediante EXP.

El resto de las constantes PI (π) se obtiene la desde la Autoguía Génova.

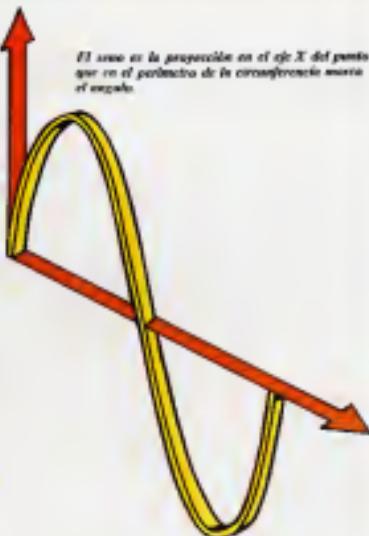


Los logaritmos en base e, se conocen como logaritmos naturales o neperianos, en memoria del matemático JOHN NEPER. El número e, de forma análoga a PI, es una constante matemática ampliamente utilizada, cuyo valor aproximado es 2.7182818. Una regla mnemotécnica muy difundida para la memorización de este constante, es añadir a 2.7, el año de la muerte de Goya (1828). Más exactamente, la definición del número e es:

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

Ahora bien, el punto que más nos interesa en esta sección, es la forma en que nuestro Spectrum es capaz de llevar a cabo esta operación; en este sentido, hemos de hacer observar que el ordenador únicamente es capaz de hallar directamente logaritmos neperianos, que se obtienen mediante la función BASIC LN. Este tipo de logaritmos son empleados bastante frecuentemente en los cálculos matemáticos de una cierta complejidad, sin embargo, para el usuario con menores aspiraciones no reportan una gran utilidad.

Seguramente, sólo deseamos saber, a qué exponente hay que elevar 3 para que dé el resultado 243, es decir, obtener el logaritmo en base 3 de 243. No debemos preocuparnos por este pequeño detalle, podremos obtener logaritmos en cualquier base, apoyándonos en la siguiente propiedad de los logaritmos:



Dadas dos bases B y B', el logaritmo en base B' de cualquier número (N), es igual al logaritmo de N en base B, dividido por el logaritmo de B' en base B:

$$\text{Log}_{B'} N = \frac{\text{Log}_B N}{\text{Log}_B B'}$$

Según esta propiedad, conociendo el logaritmo de un número en cualquier base, en nuestro caso e, podemos saber el logaritmo en cualquier otra, ampliamente dividiendo el resultado del logaritmo en la base conocida, por el logaritmo de la base en que se desea conocer. Si aplicemos esta regla a nuestro ejemplo, tendremos que:

$$\text{LN } 243 = \frac{\text{LN } 243}{\text{LN } 3} \Rightarrow \text{PRINT LN } 243 / \text{LN } 3$$

Ahora queda por hablar de un par de propiedades más de los logaritmos, que quizás nos sean de utilidad en la simplificación de algunos cálculos con estas operaciones: el logaritmo de una potencia es igual al exponente multiplicado por el logaritmo de la base de la potencia (Log<sub>B</sub> E = E \* Log<sub>B</sub> B), el logaritmo de la base logarítmica es siempre la unidad (Log<sub>B</sub> B = 1); y el logaritmo de uno es cero (Log<sub>B</sub> 1 = 0).

Finalmente, nos resta hablar de la función (EXP), que realiza la operación inversa al logaritmo (antilogaritmo). Gracias a ella, podremos obtener el número cuyo logaritmo produce determinado resultado. Utilizando esta función, podemos escri-

ber la aproximación del número e que el ordenador emplea en sus cálculos; según lo propuesto vista anteriormente, el logaritmo de la base logarítmica es siempre la unidad, por tanto el antilogaritmo de uno debe dar siempre como resultado la base logarítmica, así pues, para obtener e, base de los logaritmos neperianos, sólo debemos escribir el antilogaritmo neperiano de la unidad:

**PRINT EXP 1**

Como es lógico, los antilogaritmos que en el Spectrum se pueden obtener directamente, son los que toman como base el número e.

Para finalizar con las funciones logarítmicas, y sólo para aquéllas que tienen la especial consideración, y los suficientes conocimientos, para comprender el sistema por el cual el Spectrum calcula los logaritmos, diremos que una parte de su ROM, más concretamente dentro de la zona denominada CALCULADOR, incorpora una rutina que genera la denominada serie de polinomios de Chebyshev, gracias a lo cual, calcula las aproximaciones a las funciones SIN, ATN, LN y EXP, y por tanto a sus derivadas (COS, TAN, ASN, T y SGRT).

## FUNCIones TRIGONOMETRICAS

En este apartado, vamos a tratar tres funciones y sus correspondientes inversas. Los nombres de las funciones son SIN, COS, TAN y ASN, ACS, ATN; estas se corresponden con SENO, COSENO, TANGENTE y ARCOSENO, ARCCOSENO, ARCTANGENTE. En todos los casos, la unidad angular que se ha de utilizar para expresar el argumento de las tres primeras funciones (no de sus inversas), es el RADIAN, cuya longitud es igual al radio de la circunferencia, por tanto, los ángulos posibles en una circunferencia, expresados en radianes, osculan entre 0 y  $2\pi$ . Como la mayoría de nosotros sabremos,  $\pi$  es la constante matemática que relaciona la longitud de una circunferencia con su diámetro, y tiene el valor aproximado 3.1415927. Manejando RADIANES, se hacen muy frecuentes referencias a este valor, y el Spectrum lo tiene almacenado como una constante interna, de forma que puede ser empleada por el usuario como si se tratase de una variable de nombre PI, cuyo valor está predefinido y no es posible alterar. Así, por ejemplo, si deseamos conocer el valor que está constante

*El coseno es la proyección en el eje X del punto que en el perímetro de la circunferencia marca el ángulo*

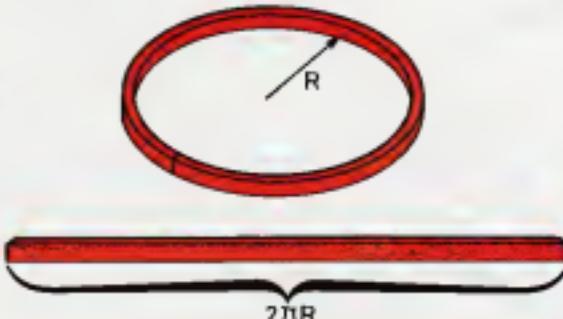


bene en el ordenador, pediremos escribir **PRINT PI**.

Deberemos tener mucho cuidado, para no confundir en el Spectrum la constante PI con una posible variable numérica PI, a efectos del ordenador, son absolutamente diferentes. La constante PI se obtiene pulsando la tecla M en el modo extendido, y corresponde al carácter 167 en el código del Spectrum, mientras que la posible variable PI, se introduce carácter a carácter, y es equivalente a p, PI o pi.

Si al encender el ordenador, le pedimos que nos diga el valor de la variable PI (**PRINT PI**) veremos que nos contesta con el error 2 Variable no

*La constante PI es la relación entre la longitud de una circunferencia y su diámetro, su valor aproximado es 3.1415927*

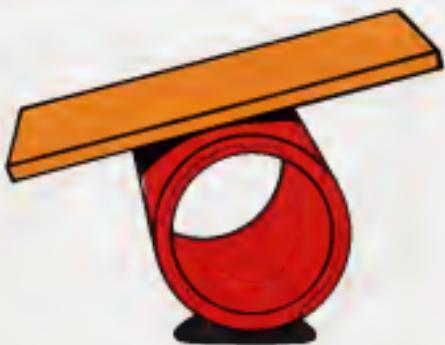


!

Las funciones inversas de las trigonométricas SIN, COS y TAN, son, respectivamente, ASN, ACS y ATN

\*

La base de los logaritmos neperianos es el número e. Podemos obtener una aproximación de el mediante de **PRINT EXP 1**



*La tangente es el cociente entre el seno y el coseno de un ángulo.*

*La función inversa del seno es el反正弦.*

found (variable no encontrada), mientras que si ejecutamos PRINT PI, siendo PI la constante (EXTENDED MODE y M), obtendremos el valor que ya conocemos.

Del mismo modo, si intentamos asignar un valor a la posible variable PI, observaremos que la tarea se lleva a cabo sin ninguna complicación, pero los valores obtenidos a continuación con PRINT PI (variable) y PRINT PI (constante), serán absolutamente diferentes (ya no ser que hayamos ejecutado LET PI=PI). Por otra parte, cualquier intento por alterar el valor de dicha constante, ya sea mediante LET, INPUT o cualquier otro sistema, será rechazado por el syntax checker.

Una buena idea, para evitar posibles confusiones en los listados, entre la variable PI y la constante PI, es escribir la variable siempre con iniciales. Hechas estas adiciones, entremos de lleno en el tema que nos ocupa.

Para hacernos una idea de lo que realmente significan las funciones trigonométricas, nos debemos imaginar una circunferencia de radio la unidad (circunferencia goniométrica), y un punto situado en su perímetro.

Esta circunferencia está seccionada por dos ejes horizontales y verticales, que pasan por su centro, dividiéndola en cuatro cuadrantes numerados del I al IV. Para su numeración, se toma como origen el punto de intersección del perímetro de la cir-

conferencia con el extremo derecho del eje horizontal, considerando por convenio el sentido de desplazamiento positivo el antihorario, es decir, el contrario al de las agujas del reloj. Del desplazamiento de este punto por el perímetro de la circunferencia, se deducen una serie de relaciones entre el ángulo descripto en el movimiento y las proyecciones sobre los ejes horizontal y vertical. Las proyecciones sobre el diámetro horizontal o eje X, corresponden a la representación de la función coseno, y las proyecciones sobre el diámetro vertical o eje Y, las del seno. Los valores del seno y del coseno en el primer cuadrante son positivos; en el segundo, el seno es positivo, mientras que el coseno es negativo; en el tercero, tanto el seno como el coseno resultan negativos; por último, en el cuarto cuadrante, el seno resulta negativo y el coseno positivo, con lo cual, a lo largo de la circunferencia, se producen todas las combinaciones posibles. Otra propiedad interesante de las funciones angulares, que se nos hace evidente dada la estructura circular en que el imaginario punto del perímetro se desplaza, es que los valores de las funciones se repiten una vez cumplida una vuelta completa, por tanto, podemos afirmar que

$$\text{SIN } X = \text{SIN } (X + N \cdot \pi) \text{ y } \text{COS } X = \text{COS } (X + N \cdot \pi)$$

Siendo X el ángulo descripto, y N cualquier número entero múltiplo de dos, que expresa el doble del número de «vueltas» dadas a la circunferencia.

Existe una tercera función trigonométrica básica, que se obtiene como la relación existente entre el seno y el coseno de un ángulo. A esta relación se le conoce con el nombre de tangente, y pode-

*Las dos unidades más fundamentalmente utilizadas en la goniometría (medición de ángulos), son el radian y el grado sexagesimal.*



*La función inversa del seno es el反正弦.*

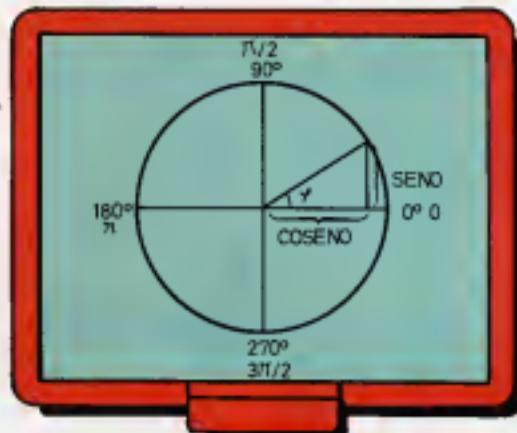


mos definirlo como  $TAN X = \frac{\text{SIN } X}{\text{COS } X}$ . Tanto la función seno como coseno, oscilan siempre entre los valores 0 y 1. Esto resulta obvio, si pensemos en que el valor mínimo que pueden adquirir es cero, como es el caso del seno de 0 o del coseno de  $\pi/2$ , y el valor máximo posible es 1, puesto que el radio de la circunferencia es la unidad, como es el caso del seno de  $\pi/2$  o el coseno de 0.

De lo dicho anteriormente, podemos deducir que cualquier valor tendrá sentido como argumento de las funciones SIN y COS, pero no sucederá así con los argumentos de la función TAN. Dado que la función tangente es el cociente de las funciones seno y coseno, debemos preocuparnos de no emplear argumentos que al hacer cero el denominador, puedan llevarnos a un error del tipo 6 Number too big (número demasiado grande). Esto se produce, lógicamente, con aquellos valores angulares cuyo coseno sea cero, es decir  $\pi/2$  y  $3\pi/2$ .

Por otro lado, podemos disponer de las funciones inversas a las anteriormente citadas, que son el arcoseno (ASIN), arccoseno (ACOS) y arccotangente (ATAN). Estas funciones nos permiten obtener, a partir del valor de un seno, un coseno o una tangente, respectivamente, el ángulo (arco) del cual provienen.

Esto implica que los argumentos de las funciones trigonométricas inversas, deberán estar siempre comprendidos entre 0 y 1, puesto que, como hemos visto anteriormente, los valores entre los cuales se mueven los resultados de las funciones SIN, COS y TAN son éstos. Si pese a esta advertencia, intentáramos efectuar una función inversa de este tipo, con argumento no comprendido entre 0 y 1, nos veríamos cara a cara con el mensaje: A invalid argument (argumento erróneo).



Una circunferencia goniométrica (de radio uno), más fácil de usar de representar las relaciones trigonométricas:

La conversión de un sistema a otro es muy sencilla. Sólo nos hace falta saber que el sistema sexagesimal divide la circunferencia en 360 grados; por tanto, existe una relación de 360 grados por cada 2 radianes, o lo que es lo mismo, 2 radianes equivalen a 180 grados. Así pues, para pasar radianes a grados, sólo hemos de multiplicarlos por 180 y dividirlos por  $\pi$ ; y para la operación contraria, es decir, convertir grados en radianes, multiplicar los grados por  $\pi$  y dividirlos por 180. El siguiente programa nos muestra cómo efectuar la conversión de una forma práctica:

```

10 REM CONVERSION ANGULAR - J.M.
20 LOPEZ MARTINEZ
20 PRINT "GRADOS", "RADIANES"
30 FOR I=0 TO 31:PRINT "-"&NEXT I
40 INPUT "Angulo:";ALINE A
50 IF A<>"G" AND A<>"R" THEN
60 TO 40
60 IF A="R" THEN GO TO 90
70 IF A<-360 OR A>360 THEN GO TO 40
80 PRINT A/180*PI: GO TO 40
90 IF A<-2*PI OR A>2*PI THEN GO TO 40
100 PRINT A/PI*180/A: GO TO 40

```

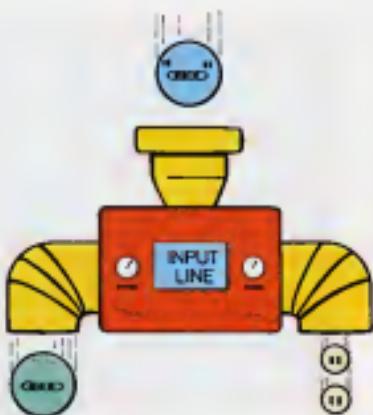
Para el manejo del programa, sólo hemos de introducir el ángulo a convertir (seguido de ENTER)

## UNIDADES ANGULARES

Cuando comenzamos a hablar de las funciones trigonométricas, aseguramos que la unidad en que se miden las magnitudes angulares es el radio, sin embargo, si bien es cierto que ésta es la unidad que más se maneja en medios técnicos, y que emplea el ordenador, la mayoría de nosotros utilizamos a tal fin el llamado sistema SEXAGESIMAL, en el cual la unidad es el grado sexagesimal, sus submúltiplos: el minuto (60 minutos=1 grado) y el segundo (60 segundos=1 minuto).



*La función inversa de la tangente es el arccotangente.*



**INPUT LINE:** supone las órdenes que acompañan la entrada de variables de cadena.

**i!**

La función logarítmica (aplicada a los signos numéricos) se obtiene en el Spectrum mediante LN.



La potenciación es la función con más prioridad dentro de las operaciones matemáticas básicas.



Los funciones BASIC SIN, COS y TAN, corresponden a las relaciones trigonométricas básicas SEÑO, COSENO y TANGEN-TE respectivamente.

RADIANES) subrayada con guiones, las líneas 70 y 90 efectúan depuraciones de los valores introducidos y, finalmente, las líneas 80 y 100, llevan a cabo el cálculo e impresión de los resultados. Ahora bien, intencionadamente hemos dejado para el final la instrucción 40, puesto que en ella aplicaremos a la entrada de datos (INPUT) algunas conceptos nuevos, que conviene analizar más detalladamente. En primer lugar, utilizaremos un INPUT en el que se visualizará el literal especificado, pero en vez de una sola variable, se producirá la entrada de dos, una numérica (A), que indica la magnitud del ángulo a convertir, y otra de cadena (U\$), que portará la unidad de medida angular (GRADOS-G, RADIANES-R).

Observemos que la primera variable del INPUT será solamente inmediatamente a continuación del mensaje de petición (para designar este tipo de mensajes se utiliza la palabra inglesa PROMPT), mientras que la segunda lo hará en la columna diecisésis, debido al separador coma (,) que se ha introducido entre las dos variables. Para la entrada de la segunda variable (U\$), del tipo cadena, hemos utilizado el formato INPUT LINE, cuyo objetivo es la supresión de los comillas que en los INPUT acompañan a los datos de este tipo. De esto deducimos, que INPUT LINE sólo puede ser empleado con variables de cadena, y que tiene una finalidad intrínsecamente estética. Por último, conviene recordar que para abandonar los INPUT de variables numéricas, emplearemos el comando STOP, y para los de variables de cadena, previamente a la realización de esta operación, procediremos a la eliminación de los comillas, que señalan este tipo de datos, bien mediante DELETE, o bien mediante EDIT. Sin embargo, en el caso del INPUT LINE ninguna de los dos sistemas funcionará, puesto que ni la variable a entrar es numérica, ni aparecen comillas que poder suprimir.

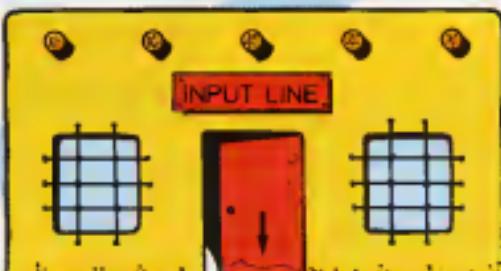
Para detectar los programas durante la ejecución de este tercer tipo de INPUT, debemos pulsar la tecla CURSOR ABAJO (CAPS SHIFT + 6), lo que producirá la generación de un mensaje del tipo R STOP en el INPUT.

y a continuación la unidad en que se encuentra, para ello, utilizaremos una R cuando sean radianes, y una G cuando sean grados sexagesimales. A este respecto, conviene advertir que dichas letras deberán entrarse en mayúsculas, para que sean reconocidas por el programa (líneas 50 y 60).

En lo referente al programa, en su mayoría se utilizan elementos que ya conocemos: la línea 10 contiene el REM de nombre y autoría, la 20 y 30 realizan la impresión de una cabecera (GRADOS

Para producir de distinción de los programas diferentes de un INPUT LINE, es necesario pulsar CURSOR ABAJO (CAPS SHIFT + 6).

El resultado visual de las siguientes líneas:



# DINAMISMO DE PANTALLAS

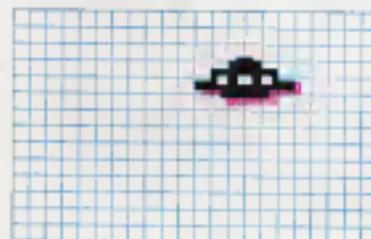


Como ya mencionamos anteriormente, la pantalla es el medio fundamental de comunicación con el usuario y, por tanto, a la hora de la realización de un programa, hemos de tener siempre en la importancia de la misma. Como vimos anteriormente, el tratamiento de la pantalla se apoya en tres puntos fundamentales:

- Dibujo de la pantalla.
- Utilización del color.
- Dinamismo.

Pero ¿qué es el dinamismo? Caeímos en la tentación del chiste fácil, y diremos que no es sólo importante cuidar la estética de la pantalla, sino también su «dinamita», esto debe pasar de ser un mero juego de pereza y convertirse en una máxima que nos acompañe durante todas nuestras tareas de programación.

El ordenador, al comando que otros vehículos de información, como el carnet, la fotografía o el libro, dispone de la gran ventaja de ser dinámico, es decir, en la mano del programador está el factor, a su gusto, el contenido informativo, y como no, la forma en que éste se presenta. Desaprovechar esta cualidad del ordenador, sería un error



En la animación de gráficos, se ejerce el efecto de desplazamiento de los sprites sobre un fondo fijo.

impardonable. Encendemos el ordenador y vamos en un ejemplo el efecto que el dinamismo puede imprimir a la simple presentación de un mensaje:

```
10 PRINT AT 11,7;"PULSA UNA TECLA"
20 IF INKEY$="" THEN GO TO 20
30 CLS
40 PRINT AT 11,7;OVER 1;"PULSA UNA
TECLA"
50 BEEP .5,60
60 IF INKEY$="" THEN GO TO 40
```

*El avión es un ejemplo clásico de dinamismo a pantalla completa, puesto que todo él (fondo y avión) se desplaza de manera uniforme.*



Evidentemente, el objetivo de la presentación del mensaje (transmitir al usuario una breve información), ha sido cumplido más ampliamente en el segundo caso, puesto que, además de esa transmisión de información, se ha reclamado previamente la atención del destinatario de la misma.

Los efectos dinámicos son, por tanto, esenciales en la programación, y con vistas a facilitar su estudio pueden ser divididos en dos grandes grupos, según la magnitud de la zona de pantalla que se vea afectada por su acción:

- Dinamismo de pantalla.
- Dinamismo de gráficos.

En el primer caso, la zona sobre la que se aplica el efecto de dinamismo es de dimensiones considerables, generalmente toda la pantalla, aunque las técnicas utilizadas a tal fin pueden ser empleadas también en áreas más restringidas

!!

Los bits dentro de un byte se numeran del 0 al 7, a partir de la derecha. El número-significativo determina la importancia que un bit tiene en el valor de un byte. Así pues, los bits serán tanto más significativos cuanta más a la izquierda se encuentren, es decir, cuanto mayor sea la posición que ocupan.

\*

Debido a la disposición de los bits dentro de los bytes del área de atributos, el valor decimal de uno de estos bytes se puede obtener por la siguiente fórmula: valor = código de fondo \* 8 + brillo \* 64 + intensidad \* 128

**i!**

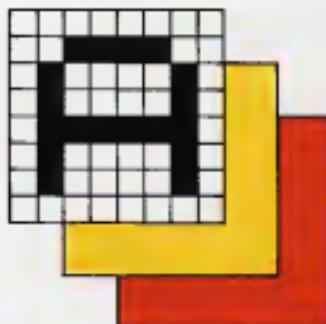
Cartela del archivo de imágenes correspondiente a un punto de la pantalla del Spectrum. Puede que cada uno de ellos sea controlable independientemente; se dice que el pixelaje de nuestro ordenador es DIRECCIONABLE PUNTO A PUNTO.



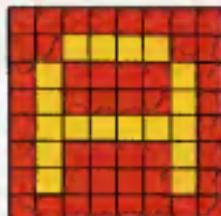
A la misma unidad gráfica que en un ordenador se puede representar se le denominan pixel. Abreviatura de la palabra inglés picture element cuyo significado es ELEMENTO GRÁFICO.



La cantidad de pixeles que se pueden representar simultáneamente en un ordenador depende de la RESOLUCIÓN de la pantalla del mismo. En el caso del Spectrum, se pueden representar 4096 pixels, por lo que se le considera de alta resolución.



En el Spectrum, se dibujen que se ven al televisor o monitor se ven de dos formas: archivo de imágenes (mapamundi de puntos) y dentro de unidades moduladas.



micas a emplear son absolutamente distintas de los utilizados para el dinamismo a pantalla completa.

Finalmente, podemos hablar del dinamismo en aplicaciones muy concretas, como es por ejemplo la presentación de mensajes, de extraordinaria importancia debido a su frecuente uso, y que disfruta del beneficio de ciertos trucos y técnicas especiales, obtenidas por extensión de los que se emplean para la animación de gráficos, y que se benefician de la mayor facilidad de tratamiento que el lenguaje BASIC aporta para los textos. Un último punto a destacar en el tema del dinamismo, es la aplicación del sonido a este fin, sin embargo, éste merece un tratamiento independiente, por lo que será excluido de las técnicas de animación general que estudiaremos en las próximas páginas.

Aunque sin duda existentes deseos de adentrarnos en las técnicas de dinamismo, con las cuales podremos obtener espectaculares efectos, deberemos frenar nuestro impetu y adquirir previamente unas breves ideas básicas sobre la forma en que nuestro Spectrum trata la información que se envía al televisor o monitor. Estos conocimientos nos ayudarán a comprender mejor algunas técnicas y trucos, que debido a la utilización del código máquina y otras herramientas de relativa complejidad, nos podrían resultar un tanto arduos o difícilmente asimilables sin base basí-

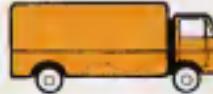
co, como, por ejemplo, un tercio o dos tercios de la pantalla. En todo caso, podemos decir que las unidades gráficas, constituyen generalmente rectángulos con el marco de la pantalla, y el efecto de dinamismo se distingue particularmente porque se aplica sobre el área completa que afecta, incluyendo tanto el fondo como el primer término, que puede estar constituido por una o varias unidades gráficas (texto, gráficos, etc.).

El segundo tipo de dinamismo, se distingue del anterior precisamente porque el objeto del mismo es la animación de determinadas unidades gráficas, generalmente reducidas en tamaño, y sobre las cuales se actúa independientemente del fondo, por ende, los sujetos pasan de su acción, que en el argot microinformático se conocen como avituales (unidades gráficas), no suelen ser superficies regulares, debido a lo cual las téc-

*El proceso de creación de imágenes en el Spectrum es llevado a cabo por varios elementos de su hardware:*



MEMORIA



U.L.A.



P.A.L.

## LA PANTALLA EN LA MEMORIA

La emisión de imagen en el Spectrum es un proceso de relativa complejidad, que involucra varios elementos del ordenador: la U.L.A se encarga de recoger la información e transmitir de la memoria R.A.M.; desde aquí, los datos son enviados al controlador P.A.L y posteriormente al modulador de U.H.F., que se encarga de preparar la señal para que sea recibida por el televisor a través del canal 3B de U.H.F. (este último paso se omite al utilizar un monitor).

Como hemos visto, muy a grandes rasgos, el proceso de emisión de imagen es automático y consiste directamente a dispositivos físicos del apartado hardware, gracias a lo cual, nuestra tarea de programación se limitará únicamente y exclusivamente a depositar en la zona de memoria adecuada, los datos que se desean transferir a la pantalla; este área de la memoria se denomina MEMORIA DE PANTALLA y ocupa un total de 6.912 bytes, comprendidos desde la dirección 16384 (primera de la memoria R.A.M.) hasta la 23295 (última inclusiva).

Para cualquier profano en la microinformática, la imagen transmitida por el ordenador es una determinada forma multiescalar, sin embargo, para el Spectrum, y por ende para el programador, la imagen se divide en dos partes claramente diferenciadas por un lado, los puntos que la componen, y por otro el color de estos. Del mismo modo, la zona de memoria de pantalla se divide en dos áreas claramente distintas, destinadas cada una de ellas a contener uno de estos elementos de la imagen (figura y color).

La primera de ellas se denomina ARCHIVO DE IMAGEN, y ocupa 6.144 bytes (desde la dirección 16384 hasta la 22527, ambas inclusivas). Esta zona se destina al almacenamiento de la configuración de la pantalla, es decir, la distribución de puntos que conforman la imagen.



MODULADOR



*El área de memoria dedicada al almacenamiento de la pantalla, se divide en dos zonas: una para imagen y otra para atributos.*

La otra zona se conoce como ARCHIVO DE ATRIBUTOS, y ocupa los últimos 768 bytes de la memoria de pantalla (desde 22528 hasta 23295), siendo su misión almacenar la información referente al color de la pantalla.

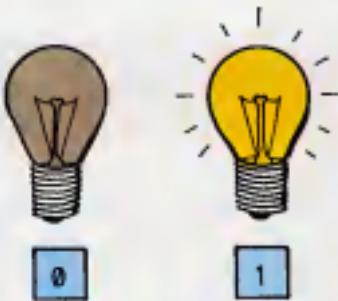
## EL ARCHIVO DE IMAGEN



Cada uno de los puntos que configuran la imagen en la pantalla, puede estar «encendido» o «apagado» (o de primer término «color de fondo»), dado que esta circunstancia supone únicamente dos posibles estados, cabe su representación informática mediante un bit, en el que se adopta el convenio 1-ENCENDIDO, 0-APAGADO. La imagen en el Spectrum está constituida por un total de 48.162 puntos, distribuidos en 192 líneas de 256 puntos cada una. Puesto que cada 8 bits asumen un byte de información, los puntos horizontales se agrupan de ocho en ocho, dando lugar a un archivo de imagen compuesto por 192 líneas de 32 bytes cada una (6.144 bytes).

Por tanto, para conectar determinado punto de la pantalla, sólo tendremos que depositar la información binaria conveniente en el byte del archivo de imagen que deseemos. Así, por ejemplo, efectuando POKE 16384, 255 conectaremos los ocho primeros puntos horizontales de la pantalla, puesto que la configuración binaria de 255 es 11111111, si sólo deseamos conectar un punto, no tendremos más que buscar la configuración

Todos los bytes que componen un programa escrito en BASIC deben comenzar por un número comprendido entre 1 y 2999. Por esto, al ejecutar por ejemplo, LIST 30000, la pantalla se ilumina y el informe OK, corroboran la imposibilidad de encontrar la línea. Sin embargo, al efectuar LIST 48162 observamos el mismo resultado que con LIST 10: es decir el ciclo del argumento del comando LIST se repite a partir de 48162, saliendo para cumplirlo con LIST. Problema: el siguiente programa 10 INPUT "LISTADO A PARTIR DE", X 20 CLR PRINT "LIST", X PRINT 30 LIST X GOTO 10

**BIT**

Este bit del archivo de imagen represente en la pantalla un punto encendido o apagado.

Cuando realizamos la impresión mediante el comando TAB, durante el salto de una columna a otra, el ordenador memoriza el color en curso en ese momento. El siguiente programa lo demuestra:

```
10 FOR A=0 TO 20
20 PRINT PAPER 7,
TAB 0,A, PAPER
RND*6 STAB 15,A,
PAPER 7,A,PAPER
RND*6 STAB 29,
PAPER 7,A
30 NEXT A
```

Si cambiamos los TAB anteriores por AT, veremos que cuando utilizamos este último no saltará la rotación.



Cuando realizamos ciertos manejos con nuestra Spectrum muchas rotaciones se presentan con gran número de decimalas, a veces, innecesarias. Si queremos limitar el número de estos, podemos incluir como subrutina en nuestros programas la línea 20 del listado siguiente:

```
10 INPUT "CUANTOS DECIMALES ",D,"NUMERO A REPRESENTAR"
20 LET A= INT (N*10^D)/10^D
30 PRINT A GOTO 10
```

A continuación, se representan las ocho primeras líneas de las filas 9 a 16, y una vez finalizado este tramo, es decir, representadas las ocho octavas líneas de la fila 9 a la 15, se pasa a la representación del último tramo, siguiendo el mismo sistema. Bueno, ¡por lo menos los tres tramos si que están por orden!

Esta organización, aparentemente un tanto caótica, se debe a razones de hardware, que en nuestro caso es como decir «razones de peso». En todo caso, la distribución de la pantalla es algo que debaremos tener muy presente al experimentar con las técnicas de dinamismo a pantalla completa. Por otra parte, ésta es la explicación de por qué las pantallas que el Spectrum almacena, se cargan de una manera tan poco peculiar.

Para dar un repaso a cómo se estructura el archivo de imagen en nuestro ordenador, ejecutemos el siguiente programa, en el que la pantalla se irá llenando de líneas según el orden de los bytes que las controlan:

```
10 PAPER 7,INK 0: BORDER 6:CLS
20 FOR I=0 TO 21
30 PRINT I
40 NEXT I
50 FOR I=16384 TO 22527
60 POKE I,255
70 NEXT I
80 GO TO 80
```

Como habremos podido observar, la diferencia más que desde el punto de vista de la programación se hace entre la zona de edición (inicialmente, las dos últimas líneas) y la de pantalla, no tiene ningún efecto en el archivo de imagen, que considera la pantalla como un todo único.

**EL ARCHIVO DE ATRIBUTOS**

binaria adecuado, por ejemplo POKE 16384,1 conecta el octavo punto de la pantalla (1 decimal = 00000001 binario), POKE 16384,128 el primero (128 decimal = 10000000 binario), o POKE 16384,197 los puntos 1, 3, 7 y 8. Habremos observado claramente, que la modificación del comando de la dirección 16384 (primera del archivo de imagen), altera lo que sería la primera fila del primer carácter de la primera fila, de forma similar, 16385 controla la primera fila del segundo carácter de la primera fila. Hasta este punto todo es muy fácil de entender, ya que cada byte se corresponde con una columna de caracteres; sin embargo, según este razonamiento, el byte 33 del archivo de imagen, debería corresponder a la segunda linea del primer carácter de la primera fila, y esto lamentablemente no se cumple: este byte es el que controla la primera linea del primer carácter de la segunda fila.

Así pues, las complicaciones aumentan, ya que en la memoria no se ordenan los bytes tal y como luego aparecen en la pantalla del televisor, sino que primero se encuentran las primeras líneas de todas las filas de caracteres, a continuación las segundas, las terceras, etc. hasta llegar a las octavas (últimas).

Lamentablemente, las desgracias nunca vienen solas, y esta distribución tampoco es del todo cierta. A efectos de estructura de la memoria, la pantalla se divide en tres tramos de ocho filas cada uno (64 bytes). Cada uno de ellos ocupa, por tanto, 2 K de memoria ( $2 \times 1024 = 2.048$  bytes), lo que suma el total de 6 K del archivo de imagen ( $6 \times 1024 = 6.144$  bytes).

Así pues, la exacta distribución de este archivo es: primeras líneas de las ocho primeras filas, segundas líneas de las ocho primeras filas, etc. hasta las octavas líneas de las ocho primeras filas,

en esta otra zona de pantalla, no encontraremos problemas de estructura. Cada byte corresponde a un carácter, y contiene la información relacionada con el mismo en lo referente al color. Su distribución está bien clara: los bytes se reparten de la misma forma que los caracteres que controlan en la pantalla, por tanto, el primer byte del archivo de atributos corresponde al primer carácter de la primera fila, el segundo al segundo, el tercero al tercer, etc.

En cuanto a la información que cada uno de estos bytes porta, se distribuye de la siguiente manera: los tres bits menos significativos (los de

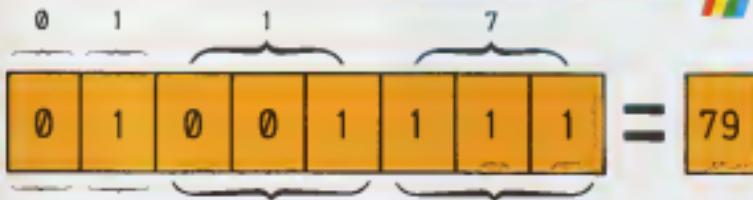


La organización del archivo de imagen donde la pantalla es una matriz cuadrangular, cada uno de los cuadros ocupa 2 K. Cada uno de estos bloques, se encuentra en un vector compuesto por 64 bytes de 32 bytes.

más a la derecha), codifican el color de la tinta, los tres siguientes (bits 3, 4 y 5) el color del papel, el sexto bit controla el brillo, y el más significativo el **FLASH**.

En el siguiente programa escribirímos un bucle infinito en la pantalla, y veremos como mediante POKEs al área de atributos, podemos alterar sus características de color, sin cambiar su configuración de puntos. Si queremos aumentar la veloci-

**Código byte del área de atributos para la información sobre tinta, fondo, brillo e intermitencia de un carácter de la pantalla**



0 → FLASH 0 1 → BRIGHT 1 1 → PAPER 1 7 → INK 7

!!

Los aficionados a las breves y sencillas aplicaciones, sin duda, las siguientes líneas:

```
10 CLS PAUSE 15
POKE 23624,0 CLS
PAPER 0 CLS
15 PAUSE 65 PA-
PER 7 POKE
23624,65 CLS
20 PRINT #1,"©
1982 Sinclair Re-
search Ltd" PAUSE
0
```

\*

En la foto de la izquierda aparece una pantallita muy alegra que demuestra el efecto de POKE sobre el área de atributos. He aquí el programa que la genera:

```
10 P 0 K E
23528+RND*768,
RND*256 GOTO 10
```

```
10 CLS PRINT TAB(10) "LET P=0
20 PRINT TAB(10) "FLASH...","BRIGHT...
"PAPE...","INK...
30 FOR I=0 TO 265
40 POKE 22528,I
50 LET C=I
60 IF C>127 THEN LET F=1:LET C=C-128
70 LET B=0 IF C<63 THEN LET B=1:LET
C=C-64
80 LET P=INT(C/8)
90 LET T=C-B*8
100 PRINT AT 4,7;P;AT 8,7;B;AT 8,7;T;AT
7,7;T
110 PAUSE 26
120 NEXT I
```



# ¡BINGO!



**L**o juego del bingo o lotería casera, ha alcanzado con el paso del tiempo un gran número de adeptos. Así como el póker, el parchís o los juegos de cartas, el bingo es uno de los pasatiempos más sueltos para entretenar tardes aburridas.

El programa que a continuación estudiaremos, gestiona la extracción de bolas de una forma totalmente aleatoria, como si se tratase de un perfecto bólido de bingo memorizando cada uno de los números para la posterior verificación de líneas o bingo respectivamente.

Aunque la mayoría de nosotros conoce la mecánica general de este juego, daremos un pequeño repaso a las reglas fundamentales: el elemento central es un bólido, dentro del cual se encuentran noventa bolas numeradas desde la uno hasta la noventa. Cada jugador dispone de uno o varios carteones con números, de forma que según se van extrayendo las bolas del bólido, se puede señalar en los mismos qué números de los presentes en nuestras tarjetas han aparecido. El jugador que primero tenga la suerte de tachar los cinco números de una misma línea de su cartón, será agraciado con el premio de LINEA [si esto ocurre en una misma jugada con otros carteones, se repartirá el premio entre los afortunados]. A partir de este momento, el premio de li-

nea ha sido repartido, y ningún beneficio se obtendrá a los próximos jugadores que obtengan líneas. El siguiente premio, y el más codiciado, es el de BINGO, destinado a aquellos jugadores que antes tachen todos los números de su cartón.

El Spectrum no genera los carteones, debido a lo cual tendremos que conseguirlas de algún antiguo juego de bingo perdido por casa, o en el peor de los casos, fabricarlos nosotros mismos. Para ello debemos seguir determinadas normas:

- Los carteones se agrupan en series de seis, que contienen el total de bolas del bólido.
- Cada cartón presenta tres filas divididas en nueve columnas; en cada una de las cuales se colocan los números del cartón correspondientes a una decena: en la primera, los números del 1 al 9, en la segunda del 10 al 19, etc. Con excepción del 90, que se añadirá en la última columna, junto con las decenas de ocho.
- En cada casilla sólo habrá un número, lo cual implica que no podrá existir más de un número perteneciente a una determinada decena en una misma línea.
- Cada columna del cartón contendrá al menos un número, y nunca tres.

En cuanto a los premios, sugerimos que la linea se pague a una tercera parte del total de cartones recogidos, y el bingo con las dos terceras partes restantes, pudiéndose utilizar como moneda, el popular «dollar lantienense», el «petrogibben» o cualquier otra moneda admisible internacionalmente, para este tipo de juegos de azar.

Si no tenemos de cartones de bingo o de alguna lotería casera, podemos fabricarlos nosotros mismos.



## EL PROGRAMA

El programa hace uso de un vector numérico de 90 elementos, definido en la línea 70, destinado al almacenamiento directo de cada uno de los distintos números contados. Este almacenamiento se realiza de la siguiente forma: cada bola extraída, se guardará en la celdilla del vector que corresponda a dicha bola. Así, si el número de-



tado es el 43, éste se almacenará en la casilla 43 del vector.

Para la verificación de líneas o bingo, nuestro Spectrum utiliza el mismo sistema. Recoge mediante un INPUT el número a verificar, recupera el contenido de la celdilla designada a ese número, y compara dicho contenido con el valor intro-

ducido. Si los dos valores son iguales, el ordenador pide el siguiente número a verificar; en caso contrario, envía un mensaje indicando que le linea o el bingo, dependiendo de la verificación que estamos efectuando, no es correcto.

La ejecución del programa puede ser interrumpida en cualquier momento, pulsando la tecla «B».

COMIENZO —————► FINA

## BASIC

## CÓDIGO MAQUINA

BINGO

El programa BINGO está compuesto por un apartado BASIC y una subrutina en código máquina, que deberán quedar almacenadas en la cinta, según el orden que se

Lo que dará acceso a un menú de tres opciones: comenzar un nuevo bingo, verificar la linea ganadora y verificar si bingo ganador.

En este programa, hemos utilizado nuevamente el subrutina en código máquina de caracteres gigantes (PSION Computer), por lo cual, aquellos que deseen usar el presente programa de bingo, deberán introducir lo citado subrutina aparcada anteriormente (programa LA BOMBA), bien a través del listado, siguiendo las instrucciones que a tal fin se brindarán en el comienzo del programa mencionado, o bien grabándolo a continuación del reporte BASIC, si es que ya fue introducido en aquel momento.

La auto ejecución del programa es obligatoria, por tanto, para la grabación del programa, tendremos que emplear el comando **SAVE "BIN-  
GO":LINE 1030**.

# SPECTRUMANIA

para Spectrum 16K ó 48K

2

II época



**RED CAR.** En la parrilla de salida parten los motores, ¿serás capaz de superar las propias marcas?

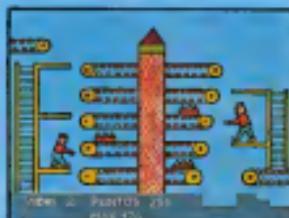


**ROBOT FACTORY:** Unos defectuosos robots-misioneros te persiguen mientras intentas cumplir tu misión.



- ESTAROS ORBULOS DE SATURNO
- UNIFORME 3
- EL DIAZO DEL MISTERIO EN EL AGUJERO KLOTHOROS

**ASTRONOMIA.** Observa, agujeros negros, nebulosas, planetas... El Universo no tiene secretos para ti.



**LOS HERMANOS MARIO.** La rapidez de reflejos es esencial para ayudar a los Mario Bros.



**PUEBLES.** Si no consigues controlar los puentes colgantes... muchos sufren las consecuencias.

- WRECKAGE PUZZLE
- TENNIS
- CUATRO EN RAYA
- SUPER-STORE
- CÓDIGO MAQUINA- QUICK SORT
- CONCURSOS
- COMENTARIOS PROGRAMAS Y PANTALLAS DE:

  - MONTY IS INNOCENT
  - STARSTRIKE
  - TIR NA NOG
  - MATCH DAY Y... MUCHO MAS

(FAESTA A LA VENTA)  
COMPRALA EN TU  
QUIOSCO HABITUAL



BOLETIN DE PEDIDO  
Avda. de Rodes, 33 - 28029 (Madrid) Tel. (34) 52 19 20.

## Datos:

- Envío a 1.º 2.º o 3.º 4.º ejemplar / N.º 1-12\* dentro de 48/72 horas - el precio de 100.- más envío
- Envío al N.º 1-12\* ejemplar / N.º 2-11\* dentro de 48/72 horas - el precio de 100.- más envío
- Envío dentro de 1 mes desde el envío del SPECTRUMANIA al punto de venta - el precio de 100.- más envío (NO SE CUMPLIRÁN LOS DÍAS DE ENVÍO SI LOS DE 3-100.- son).

ADVERTENCIA: Los precios que se indican por 5 números de SPECTRUMANIA están del 30 de julio de 1981, recibirán un 10% de descuento.

## Título:

Nombre: \_\_\_\_\_

## Apellidos:

Dirección: \_\_\_\_\_

## Teléfono:

Provincias: \_\_\_\_\_

D.P. \_\_\_\_\_

Responde con una carta la forma de pago:

- Tarjeta de crédito (sin gastos de envío)
- DNI-Facturado (100.- más gastos de envío)
- Cheque bancario (+ 100.- más gastos de envío)
- Transferencia BANCA/MANIFORT/CAMS / AMERICAM EX 46559-1-1000-1000 de banco de canarias N.º \_\_\_\_\_
- Dólares: \_\_\_\_\_

VENTAMATIC  
DE VENTA

8 VIDEOJUEGOS Y 1  
UTILIDAD EN CÓDIGO  
MAQUINA, 1 PROGRAMA  
DE GESTIÓN, 1  
PROGRAMA EDUCATIVO,  
COMENTARIOS DE SOFTWARE,  
CONCURSOS

OFERTA ESPECIAL  
VALOR POR 10 % DE DESCUENTO Y UN  
REGALO DE REGALO  
en su primera compra  
de HARWARE Y SOFTWARE  
dicho descuento no es aplicable  
a compras realizadas  
a través de SPECTRUMANIA.



## VENTAMATIC

CONTIENE  
REGALOS SORPRESA  
2 Walabolas y 200 Programas  
Busca en la portada